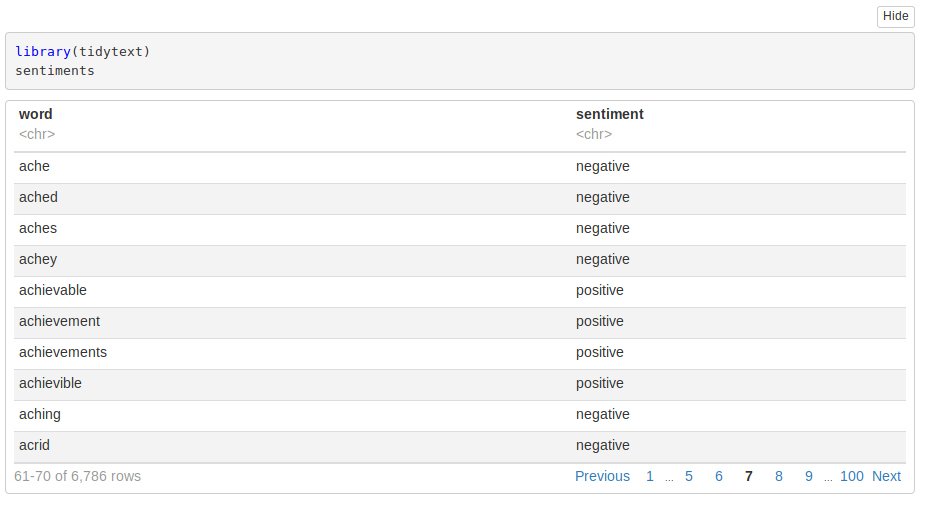
## Developing our Sentiment Analysis Model in R

The dataset that we will use will be provided by the R package ‘janeaustenR’.

In order to build our project on sentiment analysis, we will make use of the tidytext package that comprises of sentiment lexicons that are present in the dataset of ‘sentiments’.

Syntax and Output:

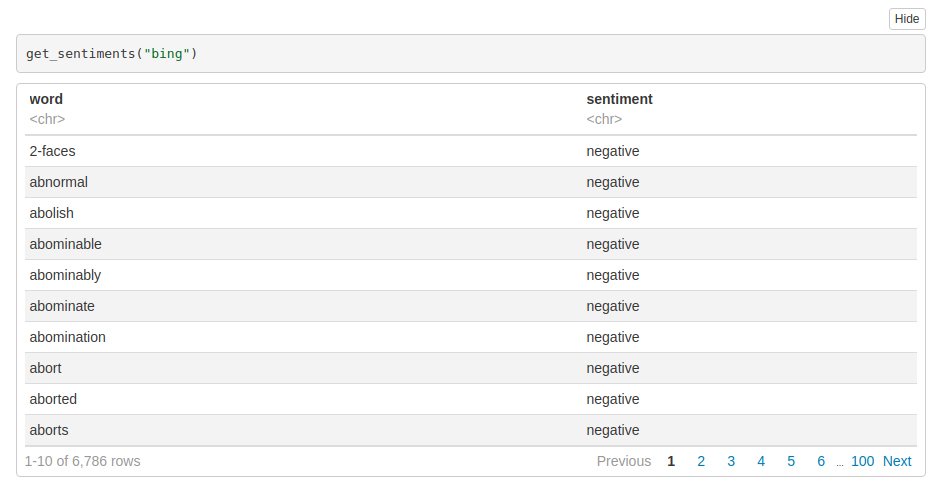


We will make use of three general purpose lexicons like –

* AFINN
* bing
* Loughran

These three lexicons make use of the unigrams. Unigrams are a type of n-gram model that consists of a sequence of 1 item, that is, a word collected from a given textual data. In the AFINN lexicon model scores the words in a range from -5 to 5. The increase in negativity corresponds the negative sentiment whereas an increase in positivity corresponds the positive one. The bing lexicon model on the other hand, classifies the sentiment into a binary category of negative or positive. And finally, the loughran model that performs analysis of the shareholder’s reports. In this project, we will make use of the bing lexicons to extract the sentiments out of our data. We can retrieve these lexicons using the get\_sentiments() function. We can implement this as follows :

Syntax and Output:



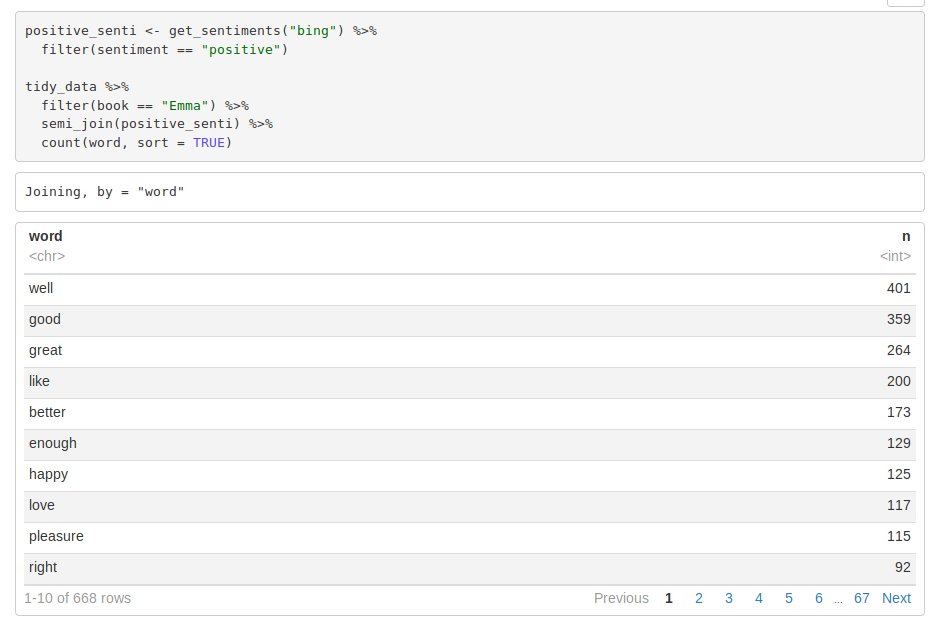
Performing Sentiment Analysis with the Inner Join

In this step, we will import our libraries ‘janeaustenr’, ‘stringr’ as well as ‘tidytext’. The janeaustenr package will provide us with the textual data in the form of books authored by the novelist [Jane Austen](https://en.wikipedia.org/wiki/Jane_Austen). Tidytext will allow us to perform efficient text analysis on our data. We will convert the text of our books into a tidy format using unnest\_tokens() function.

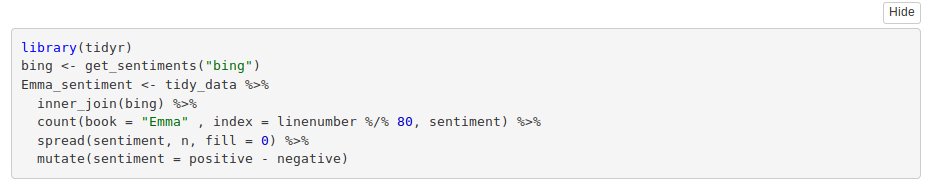
Graphical user interface, text, application, email

Description automatically generated

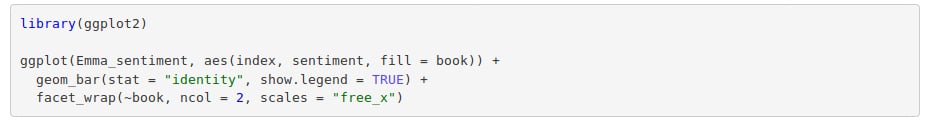
We have performed the tidy operation on our text such that each row contains a single word. We will now make use of the “bing” lexicon to filter() over the words that correspond to joy. We will use the book Sense and Sensibility and derive its words to implement out sentiment analysis model.



From our above result, we observe many positive words like “good”, “happy”, “love” etc. In the next step, we will use spread() function to segregate our data into separate columns of positive and negative sentiments. We will then use the mutate() function to calculate the total sentiment, that is, the difference between positive and negative sentiment.



In the next step, we will visualize the words present in the book “Emma” based on their corrosponding positive and negative scores.

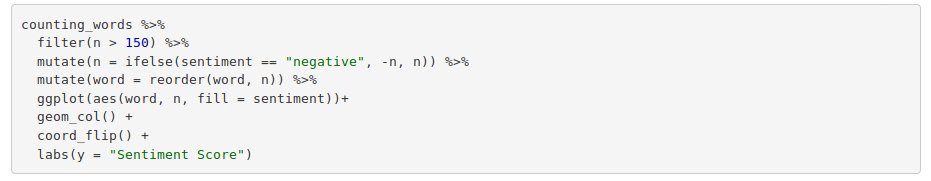


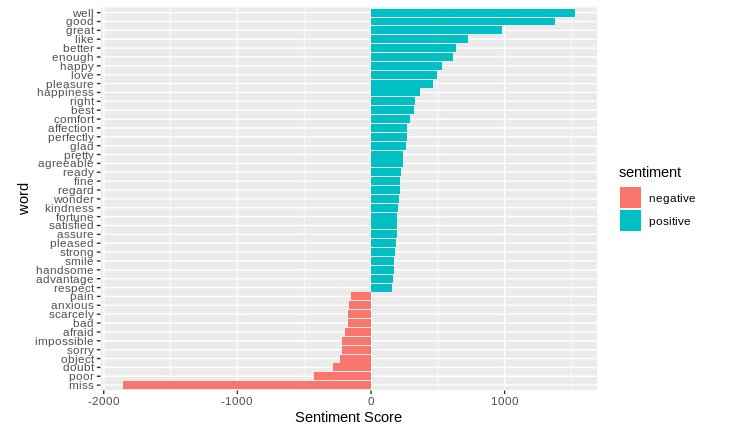


Let us now proceed towards counting the most common positive and negative words that are present in the novel.

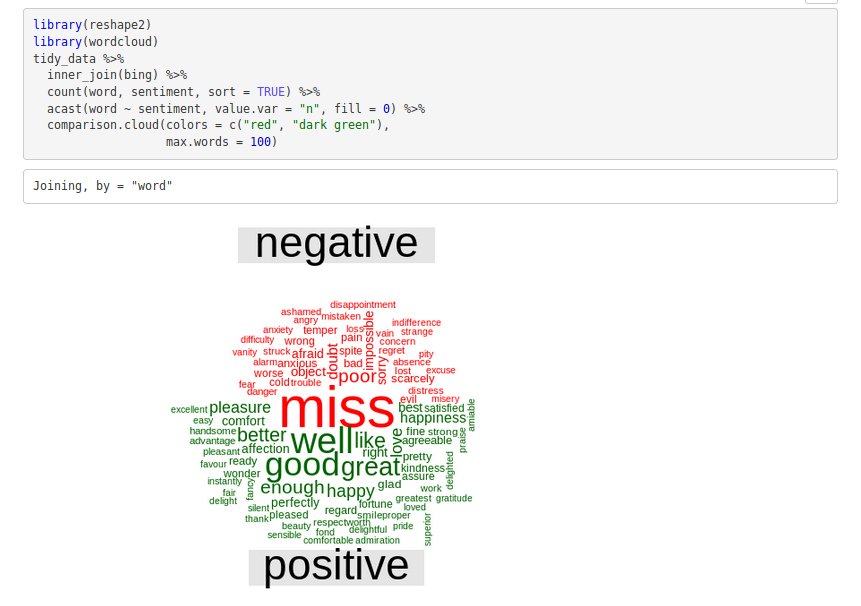


In the next step, we will perform visualization of our sentiment score. We will plot the scores along the axis that is labeled with both positive as well as negative words. We will use ggplot() function to visualize our data based on their scores.





In the final visualization, let us create a wordcloud that will delineate the most recurring positive and negative words. In particular, we will use the comparision.cloud() function to plot both negative and positive words in a single wordcloud as follows:



This word cloud will enable us to efficiently visualize the negative as well as positive groups of data. Therefore, we are now able to see the different groups of data based on their corresponding sentiments.

Summary

In this blog, we went through our project of sentiment analysis in R. We learnt about the concept of sentiment analysis and implemented it over the dataset of Jane Austen’s books. We were able to delineate it through various visualizations after we performed data wrangling on our data. We used a lexical analyzer – ‘bing’ in this instance of our project. Furthermore, we also represented the sentiment score through a plot and also made a visual report of wordcloud.